

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ВЕДЕНИЯ СТУДЕНЧЕСКИХ ЗАПИСЕЙ

Ерошевская Д. В.

*Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»
филиал «Минский радиотехнический колледж»,
г. Минск, Республика Беларусь*

*Научный руководитель: Андрейчук О.Н. – преподаватель первой категории,
магистр техники и технологии*

Аннотация. В статье рассматривается тема веб-разработки и ее основные аспекты на примере разрабатываемого веб-приложения для автоматизации ведения студенческих записей. Рассматриваются фреймворки и технологии, используемые для написания веб-приложение, а также описываются этапы процесса разработки. Также обсуждаются вопросы безопасности веб-приложений и подходы к их защите.

Ключевые слова: веб-разработка, кросс-платформенность, контейнеризация, тестирование.

Введение. В данной статье описана актуальность веб-разработки и сфера ее применения, а также преимущества использования фреймворка ASP.NET Core в качестве инструмента разработки серверной части веб-приложений. В статье описаны возможности кросс-платформенности, высокой производительности, безопасности, гибкости, облачных сервисов, встроенной поддержки кэширования, открытости, легкости, интеграции с другими технологиями, предлагаемые фреймворком ASP.NET Core.

Веб-разработка – это одна из самых актуальных технологических отраслей в наше время. И это не удивительно, ведь интернет является одним из самых быстрорастущих и масштабных рынков в мире. Одной из главных причин актуальности веб-разработки является то, что сайты и приложения стали необходимыми для большинства компаний. Без них они не могут конкурировать на рынке и предоставлять свои услуги клиентам. Кроме того, создание качественного сайта или приложения может значительно улучшить брендинг компании и повысить ее узнаваемость. В целом, актуальность веб-разработки не оспаривается. Это одна из самых важных и перспективных отраслей в современном мире, которая предоставляет бесконечные возможности для творческого развития и инноваций.

Основная часть.

Процесс разработки веб-приложения начинается с выбора инструментов разработки, анализа предметной области и определения функциональных и нефункциональных требований к приложению.

Для создания качественных и безопасных веб-приложений необходимо использовать самые современные технологии и инструменты. Для разработки серверной части веб-приложения для автоматизации ведения студенческих записей был использован фреймворк ASP.NET Core, для разработки клиентской части приложения была использована библиотека React.

ASP.NET Core – это веб-платформа с открытым исходным кодом, оптимизированная для облачных вычислений, для разработки современных веб-приложений, которые можно разрабатывать и запускать на таких операционных системах как Windows, Linux и Mac. ASP.NET Core имеет множество инструментов и библиотек для работы с базами данных, авторизации и аутентификации пользователей, а также для разработки RESTful API. ASP.NET Core также поддерживает различные способы развертывания приложений, такие как Docker и Azure.

React – это JavaScript-библиотека для создания пользовательских интерфейсов. Она была разработана Facebook и используется в таких крупных компаниях, как Instagram, Netflix, Airbnb и другие.

Одной из главных особенностей React является использование виртуального DOM. Виртуальный DOM – это копия реального DOM, которая находится в памяти компьютера. React использует виртуальный DOM для ускорения работы приложения и минимизации количества обращений к реальному DOM.

React также использует компонентный подход к разработке интерфейсов. Компонент – это независимый элемент пользовательского интерфейса, который может быть использован в любом месте приложения. Это позволяет значительно сократить время разработки и повысить эффективность работы.

React имеет множество инструментов для работы с состоянием компонентов. React позволяет легко управлять состоянием компонентов и обновлять их при изменении данных.

React также имеет множество инструментов для работы с маршрутизацией и управлением состоянием приложения. Например, React Router – это библиотека для управления маршрутизацией в React-приложениях. Redux – это библиотека для управления состоянием приложения в React.

Сочетание ASP.NET Core и React позволяет создавать высокопроизводительные и масштабируемые приложения, которые могут работать на любых устройствах и платформах. Например, можно создать SPA-приложение (Single Page Application), которое будет общаться с сервером через RESTful API, используя ASP.NET Core, и использовать React для создания пользовательского интерфейса. ASP.NET Core обеспечивает высокую производительность и масштабируемость, а React обеспечивает быстрое и отзывчивое пользовательское интерфейсы.

После того как были сформулированы требования к разрабатываемому приложению и были выбраны инструменты разработки начинается написание кода и тестирование приложения. Модульное тестирование, интеграционное тестирование и тестирование производительности – это основные методы тестирования, которые могут быть использованы для проверки правильности работы приложения.

Модульное тестирование – это процесс тестирования отдельных модулей приложения, таких как классы и методы. В ASP.NET Core это можно сделать с помощью фреймворка xUnit. Фреймворк xUnit позволяет создавать модульные тесты, которые проверяют правильность работы отдельных модулей приложения.

Интеграционное тестирование – это процесс тестирования приложения в целом, проверка взаимодействия между различными компонентами приложения. В ASP.NET Core это можно сделать с помощью фреймворка Selenium. Фреймворк Selenium позволяет создавать автоматические тесты, которые проверяют работу приложения в браузере.

Тестирование производительности – это процесс тестирования приложения на его способность обрабатывать большое количество запросов и пользователей. В ASP.NET Core это можно сделать с помощью фреймворка Apache JMeter. Фреймворк Apache JMeter позволяет создавать нагрузочные тесты, которые проверяют производительность приложения при большой нагрузке.

Завершающим и критически важным шагом разработки веб-приложений является развертывание веб-приложения в рабочей среде: на гостевом хостинге или сервере. Для упрощения процесса развертывания веб-приложений используются инструменты веб-развертывания, который значительно упрощает миграцию, управление и развертывание веб-приложений. Одним из таких инструментов является Docker.

Docker – это популярная платформа, в основе которой лежит технология контейнеризации. Docker позволяет запускать Docker-контейнеры с приложениями из заранее заготовленных шаблонов – Docker-образов, Docker позволяет управлять и развертывать контейнеры на любой системе.

Контейнеризация – это процесс упаковки приложения и его зависимостей в изолированный контейнер, который может быть запущен на любой платформе. Docker – это инструмент для контейнеризации, который позволяет разработчикам создавать, управлять и развертывать контейнеры на любой системе. Простыми словами контейнер – это некая изолиро-

ванная песочница для запуска ваших приложений. Контейнеризация позволяет упростить процесс развертывания приложений, уменьшить время, затрачиваемое на настройку окружения и улучшить безопасность. Каждый контейнер имеет свою собственную изолированную файловую систему, сетевое пространство и процессы, что предотвращает конфликты между различными приложениями.

Docker образ – это шаблон (физически – исполняемый пакет), из которого создаются Docker-контейнеры. Образ хранит в себе всё необходимое для запуска приложения, помещенного в контейнер: код, среду выполнения, библиотеки, переменные окружения и конфигурационные файлы.

При запуске контейнера Docker распаковывает образ контейнера с нужными файлами и метаданными, инициализирует сеть и файловую систему и запускает приложение в соответствии с инструкциями, заданными в образе.

Docker также предоставляет множество инструментов для управления контейнерами, таких как Docker Compose, Kubernetes и Swarm. Эти инструменты позволяют управлять несколькими контейнерами одновременно и автоматически масштабировать приложение в зависимости от нагрузки.

Контейнеризация с использованием Docker является одним из самых эффективных способов развертывания веб-приложений. Это позволяет упростить процесс развертывания, улучшить безопасность и сделать веб-приложение более гибким и портативным.

Вместе с удобством использования, веб-приложения также несут в себе риски безопасности. Хакеры могут использовать различные методы для взлома системы и получения доступа к конфиденциальным данным. Поэтому, безопасность веб-приложений является критически важной проблемой.

Рассмотрим некоторые подходы к защите веб-приложений на основе ASP.NET Core.

1. Использование протокола HTTPS. HTTPS – это протокол безопасной передачи данных в Интернете. Он использует шифрование для защиты данных, передаваемых между клиентом и сервером. Использование HTTPS является одним из наиболее эффективных способов защиты веб-приложений от атак типа "человек посередине" и других видов кибератак.

2. Защита от CSRF атак. CSRF (Cross-Site Request Forgery) – это атака, при которой злоумышленник отправляет запрос от имени авторизованного пользователя без его согласия. Для защиты от CSRF атак можно использовать токены запросов (Request Tokens). Токен запроса добавляется к каждому запросу, отправляемому на сервер, и проверяется на стороне сервера. Если токен не совпадает с ожидаемым значением, то запрос отклоняется.

3. Защита от XSS атак. XSS (Cross-Site Scripting) – это атака, при которой злоумышленник внедряет вредоносный код на страницу, который может быть выполнен в браузере пользователя. Для защиты от XSS атак можно использовать HTML-кодирование. HTML-кодирование позволяет преобразовать специальные символы в их эквиваленты HTML, которые не могут быть интерпретированы как код.

4. Защита от инъекций SQL. SQL-инъекция – это атака, при которой злоумышленник внедряет SQL-код в запрос, отправляемый на сервер. Для защиты от SQL-инъекций можно использовать параметризованные запросы. Параметризованные запросы позволяют передавать параметры в запрос, которые автоматически кодируются, что предотвращает возможность внедрения вредоносного SQL-кода.

Также осуществляется защита на основе правил доступа. Аутентификация и авторизация – это процессы, позволяющие проверить легитимность пользователя и предоставить ему доступ к определенным ресурсам. Этот подход заключается в том, что пользователи могут получать доступ только к определенным страницам и функциям, которые соответствуют их ролям или привилегиям. Таким образом, управление доступом помогает предотвратить несанкционированный доступ к конфиденциальной информации. Управление доступом на основе ролей позволяет предоставлять доступ только к определенным ресурсам в зависимости от роли пользователя.

Защита веб-приложений – это сложный процесс, который требует внимания к деталям и постоянного обновления методов защиты. Использование протокола HTTPS, защита от

CSRF и XSS атак, параметризованные запросы, двухфакторная аутентификация и управление доступом на основе ролей – это лишь некоторые из методов, которые можно использовать для защиты веб-приложений на основе ASP.NET Core.

Заключение. В ходе написания статьи была изучена заданная предметная область, были получены более глубокие знания о процессе разработки веб-приложений, разворачивания веб-приложений в рабочем окружении и способах защиты от кибератак. Веб-разработка это быстро развивающаяся и динамичная сфера, которая требует от разработчиков постоянного повышения уровня знаний и постоянного обучения и овладения новыми технологиями.

Список литературы

1. Квинт, Игорь. *Создаем сайты с помощью HTML, XHTML и CSS на 100%*. – / И. Квинт. – 2-е издание, СПб.: Питер, 2012 – 448 с.
2. Статьи на Комьюнити [Электронный ресурс]. – Режим доступа: <https://timeweb.com> – Дата доступа: 25.03.2023.

UDC 004.75

CRYPTOCURRENCY, DEVELOPMENT AND APPLICATION

Eroshevskaya D. V.

*Educational institution "Belarusian State University of Informatics and Radioelectronics"
branch "Minsk Radio Engineering College", Minsk, Republic of Belarus*

Scientific supervisor: Andreichuk A.O. – teacher of the first category, master of engineering and technology

Annotation. The article discusses the topic of web development and its main aspects on the example of a web-application for automating student records management. The article discusses the frameworks and technologies that have been used to write a web application and the stages of the development process. Security issues of web applications and approaches for their protection are also discussed.

Keywords: web development, cross-platform, containerization, testing.